

# Data Structures Lab: Viva Questions and Answers

## Experiment 1: Array Operations

Q: What is an array?

A: A collection of similar elements stored in contiguous memory locations.

Q: How are array elements accessed?

A: By using their index starting from 0.

Q: What is the time complexity to access an element?

A: O(1).

Q: How do you declare an array in C?

A: int arr[10];

Q: What is the default index of the first element?

A: Zero (0).

Q: Can arrays store different data types?

A: No, arrays store elements of the same data type.

Q: What is the size of int arr[5]?

A: It can hold 5 integer elements.

Q: How do you find the length of an array?

A: By dividing total size by size of one element.

Q: What is array traversal?

A: Accessing each element of an array sequentially.

Q: Can array size be changed during execution?

A: No, array size is fixed once declared.

## Experiment 2: Stack Implementation

Q: What is a stack?

A: A linear data structure following LIFO order.

Q: What does LIFO mean?

A: Last In First Out.

Q: What are stack operations?

A: Push, Pop, Peek, and Display.

Q: Which function inserts an element in stack?

A: Push().

Q: Which function removes the top element?

A: Pop().

Q: When is a stack overflow?

A: When trying to push an element into a full stack.

Q: When is a stack underflow?

A: When trying to pop from an empty stack.

Q: How is a stack implemented?

A: Using arrays or linked lists.

Q: What is the time complexity of push and pop?

A: Both are O(1).

Q: Name a real-life example of stack.

A: Undo operation in editors.

## Experiment 3: Queue Implementation

Q: What is a queue?

A: A linear data structure following FIFO order.

Q: What does FIFO stand for?

A: First In First Out.

Q: What are basic queue operations?

A: Enqueue and Dequeue.

Q: Which operation adds an element?

A: Enqueue.

Q: Which operation removes an element?

A: Dequeue.

Q: When is queue overflow?

A: When adding to a full queue.

Q: When is queue underflow?

A: When removing from an empty queue.

Q: What are types of queues?

A: Simple, Circular, Priority, Deque.

Q: How can a queue be implemented?

A: Using arrays or linked lists.

Q: Give an example of queue usage.

A: Printer job scheduling.

## Experiment 4: Linked List Operations

Q: What is a linked list?

A: A collection of nodes connected by links.

Q: What does each node contain?

A: Data and a pointer to the next node.

Q: How is linked list different from array?

A: Array uses contiguous memory, linked list doesn't.

Q: What is the first node called?

A: Head node.

Q: What is the last node pointing to?

A: NULL.

Q: What is insertion in a linked list?

A: Adding a new node to the list.

Q: What is deletion in a linked list?

A: Removing a node from the list.

Q: What is traversal?

A: Visiting each node sequentially.

Q: How do you represent an empty list?

A: Head pointer is NULL.

Q: Name types of linked lists.

A: Singly, Doubly, Circular.

## Experiment 5: Circular Linked List

Q: What is a circular linked list?

A: The last node points to the first node.

Q: What is the advantage of circular linked list?

A: Traversal can continue indefinitely.

Q: What is the last node of circular list point to?

A: The head node.

Q: Can a circular list be singly linked?

A: Yes.

Q: Can a circular list be doubly linked?

A: Yes.

Q: How do you detect the end of circular list?

A: When the next pointer becomes head again.

Q: What is used to traverse circular list?

A: A do-while loop.

Q: Is it possible to insert at beginning?

A: Yes, by adjusting last node's pointer.

Q: Is it possible to insert at end?

A: Yes, before the head node.

Q: Where is circular list used?

A: In round robin scheduling.

## Experiment 6: Doubly Linked List

Q: What is a doubly linked list?

A: Each node has two pointers, next and previous.

Q: What does prev pointer store?

A: Address of previous node.

Q: What is the advantage of DLL?

A: Traversal in both directions.

Q: How is insertion done?

A: By updating both next and prev pointers.

Q: How is deletion done?

A: By reconnecting adjacent nodes.

Q: What is the first node called?

A: Head.

Q: What is the last node called?

A: Tail.

Q: Can we traverse backward?

A: Yes, using prev pointers.

Q: Is DLL memory efficient?

A: No, extra pointer requires more memory.

Q: Name one use of DLL.

A: In browser history navigation.

## Experiment 7: Tree

Q: What is a tree?

A: A hierarchical data structure of nodes.

Q: What is a root node?

A: The topmost node in a tree.

Q: What are child nodes?

A: Nodes directly connected below a parent.

Q: What is a leaf node?

A: Node with no children.

Q: What is a subtree?

A: Child node and its descendants.

Q: What is tree traversal?

A: Visiting all nodes systematically.

Q: Name types of tree traversal.

A: Inorder, Preorder, Postorder.

Q: What is depth of a node?

A: Distance from root node.

Q: What is height of a tree?

A: Longest path from root to leaf.

Q: Give an application of tree.

A: In file systems.

## Experiment 8: Binary Search Tree

Q: What is BST?

A: A binary tree where left < root < right.

Q: What is the key property of BST?

A: Left child < parent < right child.

Q: What is inorder traversal of BST?

A: It gives sorted order of elements.

Q: What operation inserts a node?

A: Insert().

Q: What operation deletes a node?

A: Delete().

Q: How do you search an element?

A: By comparing recursively with nodes.

Q: What is time complexity of search?

A: O(log n) in balanced BST.

Q: What causes imbalance?

A: Uneven insertions.

Q: How to balance BST?

A: Using AVL or Red-Black Trees.

Q: Where is BST used?

A: In database indexing.

## Experiment 9: Graph

Q: What is a graph?

A: A set of vertices and edges.

Q: What connects two vertices?

A: An edge.

Q: What are types of graphs?

A: Directed and Undirected.

Q: What is a weighted graph?

A: Edges have weights or costs.

Q: What is degree of a vertex?

A: Number of connected edges.

Q: What is adjacency matrix?

A: 2D array representing edges.

Q: What is BFS?

A: Breadth First Search.

Q: What is DFS?

A: Depth First Search.

Q: What is a cycle?

A: Path that starts and ends at same vertex.

Q: Where are graphs used?

A: In social networks and maps.

## Experiment 10: Sorting and Searching

Q: What is sorting?

A: Arranging data in specific order.

Q: What is searching?

A: Finding an element in data.

Q: Name sorting algorithms.

A: Bubble, Insertion, Selection, Quick, Merge.

Q: Name searching algorithms.

A: Linear search and Binary search.

Q: What is time complexity of linear search?

A:  $O(n)$ .

Q: What is time complexity of binary search?

A:  $O(\log n)$ .

Q: Which search requires sorted data?

A: Binary search.

Q: Which sorting is fastest?

A: Depends on data, often QuickSort.

Q: What is stability in sorting?

A: Order of equal elements remains same.

Q: Where is searching used?

A: In databases and lookup operations.

## Basic C Programming Questions (Extra 20)

Q: What is C language?

A: A general-purpose structured programming language.

Q: Who developed C?

A: Dennis Ritchie.

Q: In which year C was developed?

A: 1972.

Q: What is a compiler?

A: A program that converts source code to machine code.

Q: What is a variable?

A: A named location in memory to store data.

Q: What is the size of int in C?

A: Typically 4 bytes.

Q: What is printf used for?

A: To display output on screen.

Q: What is scanf used for?

A: To take input from user.

Q: What is a function?

A: A block of code that performs a specific task.

Q: What is recursion?

A: Function calling itself.

Q: What is the purpose of main()?

A: It is the entry point of program.

Q: What is syntax error?

A: Error in program rules or grammar.

Q: What is a pointer?

A: A variable that stores address of another variable.

Q: What is NULL pointer?

A: Pointer that points to nothing.

Q: What is an array?

A: Collection of elements of same type.

Q: What is a string?

A: Array of characters ending with null character.

Q: What is difference between `++i` and `i++`?

A: `++i` increments first, `i++` increments later.

Q: What is `sizeof` operator?

A: Used to find memory size of variable or type.

Q: What is a loop?

A: Repeats a block of code multiple times.

Q: What is the keyword used to declare constants?

A: `const`.